

Esame di Calcolatori Elettronici T

7 Settembre 2016 (Ing. Informatica)

Esercizio 1

Progettare un sistema basato sul microprocessore DLX dotato di **512 MB di EPROM** mappata negli indirizzi bassi e **256 MB di RAM** mappata negli indirizzi alti. Nel sistema sono presenti **tre periferiche a 8 bit**, già progettate, denominate **INPUT_PORT_1** e **INPUT_PORT_2**, in grado di ricevere dati dall'esterno utilizzando il protocollo di *handshake*, e **OUTPUT_PORT**, in grado di inviare dati all'esterno utilizzando il protocollo di *handshake*.

A ogni interrupt generato dalle periferiche è assegnata la seguente priorità statica: **INPUT_PORT_1** (più prioritario), **INPUT_PORT_2** (priorità intermedia) e **OUTPUT_PORT** (meno prioritario). Il dato letto dalle periferiche in input dovrà essere memorizzato in R12 mentre la porta in output dovrà ricevere il contenuto di R13.

Inoltre, al termine di ogni ciclo di scrittura in **OUTPUT_PORT**, dovrà essere invertito, in modo automatico, lo stato di un segnale denominato **LED_OUTPUT** (all'avvio del sistema tale segnale deve essere non asserito).

- Progettare il sistema escogitando un metodo che consenta di poter leggere in modo affidabile lo stato dei segnali di interrupt generati dalle periferiche
- Progettare la rete che consente di generare il segnale **LED_OUTPUT**
- Indicare i segnali di decodifica di tutte le periferiche, memorie e di eventuali altri segnali presenti nel sistema
- Scrivere il codice dell'*interrupt handler*, **commentando in modo chiaro ogni istruzione**, che consente di gestire le priorità indicate in precedenza. Si assuma che i registri da R20 a R30 possano essere utilizzati senza la necessità di dover preservare il loro contenuto

Esercizio 2

Descrivere l'organizzazione della memoria e i segnali emessi dal processore nel caso di un sistema con bus dati a 64 bit.

Esercizio 3

Descrivere le tipologie di interrupt presenti in un sistema a microprocessore.

TAPPING MEMOIRE

EPROM 512 Kib : $0 \times 00000000 \rightarrow 0 \times 1FFF FFFF$

RAM 128 Kib L : $0 \times F0000000 \rightarrow 0 \times F7FF FFFF$

RAM 128 Kib H : $0 \times F8000000 \rightarrow 0 \times FFFF FFFF$

CHIP SELECT

CS_RAM_L_0 = BA31 BA27 B6 0

CS_RAM_L_1 = BA31 BA27 B6 1

CS_RAM_L_2 = BA31 BA27 B6 2

CS_RAM_L_3 = BA31 BA27 B6 3

CS_RAM_H_0 = BA31 BA27 B6 0

CS_RAM_H_1 = BA31 BA27 B6 1

CS_RAM_H_2 = BA31 BA27 B6 2

CS_RAM_H_3 = BA31 BA27 B6 3

CS_EPROM_0 = BA31 BA30 B6 0

CS_EPROM_1 = BA31 BA30 B6 1

CS_EPROM_2 = BA31 BA30 B6 2

CS_EPROM_3 = BA31 BA30 B6 3

CS_INPUT_1 = BA31 BA30 B6 0

CS_INPUT_2 = BA31 BA30 B6 1

CS_OUTPUT = BA31 BA30 B6 2

CS_READ_INT = BA31 BA30 B6 3 + MEHRD

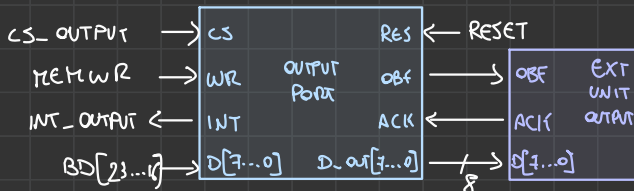
PORTES D' I/O



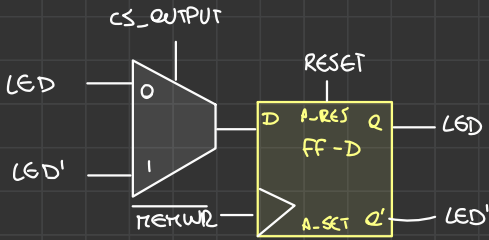
INT (TO DLX) = INT_INPUT_1 +

INT_INPUT_2 +

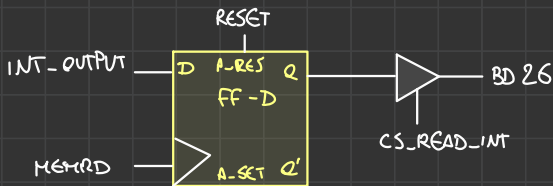
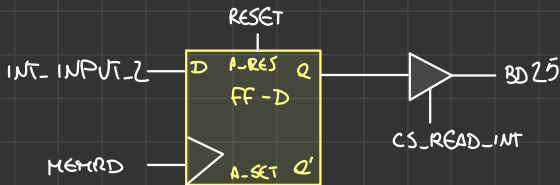
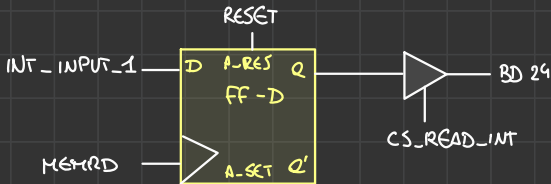
INT_INPUT_3



LED



Rete per gestire le priorità



CODICE

```

0h  LHI    R20, 0x4000
6h  LBU    R21, 0x0003(R20)
8h  ANDI   R26, R21, 0x0001
Ch  BNEZ   R26, INPUT_1
10h  ANDI   R26, R21, 0x0002
14h  BNEZ   R26, INPUT_2
18h  OUTPUT: SB    R13, 0x0002(R20)
1Ch  RFE

```

```

20h  INPUT_1: LBU R12, 0x0000(R20)
24h                      RFE;
28h  INPUT_2: LBU R12, 0x0001(R20)
2Ch                      RFE;

```

INPUT_1 = 10h

INPUT_2 = 18h

